Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete

Patrick Maier

MPI–I–2003–2–001 March 2003

Author's Address

Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
Phone: +49 681 9325-228
Fax: +49 681 9325-299
Email: `maier@mpi-sb.mpg.de`

Publication Notes

This report is the extended version of a paper presented at *Foundations of Software Science and Computation Structures* (FOSSACS 2003), a member conference of the *European Joint Conferences on Theory and Practice of Software* (ETAPS 2003), Warsaw, Poland, April 7 to 11, 2003.

Acknowledgements

The author thanks Viorica Sofronie-Stokkermans and Andreas Podelski for helpful discussions and comments and Carsten Sinz for support with theorem provers.

Abstract

Circular assume-guarantee reasoning is used for the compositional verification of concurrent systems. Its soundness has been studied in depth, perhaps because circularity makes it anything but obvious. In this paper, we investigate completeness. We show that compositional circular assume-guarantee rules cannot be both sound and complete.

Keywords

Compositional Verification, Assume-Guarantee Reasoning, Completeness

1 Introduction

The goal in compositional verification of concurrent systems is to prove that a complex system, a parallel composition of several subsystems, satisfies a complex property, a conjunction of several simpler properties. In principle, verification tools can attack such a goal directly, at least if the complex system is finite still. However, the state space of the system may be exponentially larger than that of any subsystem — a phenomenon called *state explosion* — which may cause verification to become intractable in practice. Compositional verification tries to use the modular structure of complex systems and properties to decompose intractable verification tasks into a bunch of smaller, hopefully tractable subtasks; ideally, each subtask only establishes properties of a single subsystem in isolation. Later, one deduces from all these subtasks via a suitable proof rule that the original complex system satisfies the desired complex property.

Systems, Properties. We model systems and properties uniformly as elements of $\mathbf{S} = \langle S, \wedge, 1, \leq \rangle$, a meet-semilattice with one, i.e., a partial order $\langle S, \leq \rangle$ with greatest element 1 in which the greatest lower bound $x \wedge y$ of any two elements x and y exists. In this model, the expression $x \leq y$ can have three different readings depending on whether x and y denote systems or properties, respectively. If both are systems then $x \leq y$ means that x refines y, if both are properties then it means that x entails y, and if x is a system and y a property then $x \leq y$ expresses that x satisfies y. Likewise, $x \wedge y$ denotes composition if x and y are systems, it denotes conjunction if x and y are properties, and if x is system and y a property then $x \wedge y$ — x constrained by y — is the coarsest refinement of x that satisfies y. Thus, all we require of systems (properties) is that refinement (entailment) is an order and that composition (conjunction) is an associative commutative and idempotent operation which respects the order. Section 4 will show that this abstract algebraic setting already suffices for proving incompleteness of circular assume-guarantee reasoning. In particular, no notion of computation is required, unlike in the proofs for soundness.

Example 1. Meet-semilattices are a natural model for systems and properties. For instance, in (linear-time) temporal verification, often one views systems and properties as languages over some non-empty (and possibly infinite) alphabet Σ. In this setting, refinement and entailment correspond to language inclusion, and composition and conjunction correspond to language intersection, so we have a meet-semilattice structure. How that meet-semilattice actually looks like, depends on the type of properties we want to verify.

By the characterization in [3], safety properties are expressible as prefix-closed $*$-languages. I.e., a safety property may be viewed as a subset L of

Σ^* such that for all $w \in \Sigma^*$, if w belongs to L then all prefixes of w belong to L, too. So for verification of safety properties, the meet-semilattice is the set of prefix-closed $*$-languages (over Σ). Note that these are exactly the languages generated by (possibly infinite) labeled state transition graphs, which are a natural representation of systems. In general, when verifying arbitrary temporal properties, the meet-semilattice will be the set of ω-languages (over Σ), i. e., the power set of Σ^ω. Here, natural representations of systems and properties are some more elaborate variants of state transition graphs, for instance fair transition systems [15] or (possibly infinite state) ω-automata [21]. $\square$

Proof Rules. In general, there are two kinds of proof rules for compositional verification, non-circular and circular ones. We show some examples to demonstrate the difference. Let $s_1, s_2 \in S$ be systems and let $p_1, p_2 \in S$ be properties and suppose we want to verify that the composition of s_1 and s_2 satisfies the conjunction of p_1 and p_2. (1) shows two non-circular rules for this purpose. Both rules decompose the goal into two subgoals, where the subgoals of the first rule state that system s_i satisfies property p_i, or in other words: s_i guarantees p_i. The second rule differs only in the second subgoal, which states that s_2 constrained by p_1 satisfies p_2, or in other words: if p_1 is assumed then s_2 guarantees p_2 — hence the wide-spread term *assume-guarantee rule*.

$$\frac{s_1 \leq p_1 \qquad s_2 \leq p_2}{s_1 \wedge s_2 \leq p_1 \wedge p_2} \qquad\qquad \frac{s_1 \leq p_1 \qquad p_1 \wedge s_2 \leq p_2}{s_1 \wedge s_2 \leq p_1 \wedge p_2} \tag{1}$$

Going one step further and also introducing an assumption in the first subgoal, we obtain the circular rule (2).

$$\frac{p_2 \wedge s_1 \leq p_1 \qquad p_1 \wedge s_2 \leq p_2}{s_1 \wedge s_2 \leq p_1 \wedge p_2} \tag{2}$$

Unlike the non-circular rules, (2) is unsound; for instance, if both systems are 1, the greatest element in $\mathbf{S}$, and both properties are equal and different from 1 then both premises hold but the conclusion does not. As soundness is indispensable, rule (2) must be restricted by a side condition which excludes cases as the one above. Such circularity-breaking side conditions do exist; in fact, quite a number of restricted variants of (2) are proven sound (by induction usually) in the literature, see [1, 4, 13, 18, 22] to name just a few.

Completeness. As there are many variants of sound circular assume-guarantee rules, the question arises whether some are better than others. An important criterion for rating rules is the restrictiveness of the side condition; if the side condition is overly restrictive the rule is applicable to few cases only, hence it is considered worse than a variant with a less restrictive side condition (as long as that variant is sound still). In the best case, the rule is complete, i. e., the side condition is true whenever premises and conclusion are true. Thus, the side condition of a complete rule does not restrict the rule unnecessarily since it is true whenever the rule should be applicable. Note however, that the side condition is not redundant in complete rules; it may still be indispensable for proving soundness.

Compositionality. Recall that compositional verification seeks to reduce a large, intractable goal into many smaller subgoals. The rules in (1) and (2) support this approach as the premises of these rules are less complex than their conclusions. In particular, no premise involves the composition of the systems s_1 and s_2 any more, so verification of the subgoals is more likely to be tractable than direct verification of the goal. However, when using a circular rule which is restricted by a side condition, it does not suffice to verify the subgoals that arise from the premises; additionally, we need to prove that the side condition holds. It may be the case that this proof requires to consider both systems simultaneously — e. g., for establishing some mutual exclusion property — and thus involves some aspects of the composition, which is against the spirit of compositional verification. Therefore, a rule can be called *compositional* only if checking the side condition is possible without taking into account both systems simultaneously, i. e., only if the side condition is expressible as a boolean combination of sub-conditions, each of which involves at most one of the systems.

Plan. Section 2 formally presents proof rules which are restricted by a side condition and defines soundness and completeness. Section 3 specifies what we mean by circular assume-guarantee reasoning in the context of compositional verification and formalizes the precise requirements for rules to be compositional. Section 4 proves the main result that compositional circular assume-guarantee rules cannot be both sound and complete. Finally, Section 5 discusses related work and Section 6 concludes.

2 Inference Rules

Terms, Formulas. We fix a set of variables *Var*. Terms are built inductively from variables in *Var*, the nullary operator $\top$, called *top*, and the binary operator $\sqcap$, called *meet*. We consider top neutral w. r. t. meet, which is seen as associative, commutative and idempotent. We call a term t *atomic* iff t is a variable or t is top. By $var(t)$, we denote the set of variables occuring in t, and we say that a term t' is a *subterm* of t iff $var(t') \subseteq var(t)$.

A formula φ is a pair $\langle t, t' \rangle$ of terms, written as $t \sqsubseteq t'$. We refer to t as the *left-*, to t' as the *right-hand side* of φ. We call a formula *trivial* iff its right-hand side is $\top$. We denote the set of variables occuring in φ by $var(\varphi)$,

i. e., $var(\varphi) = var(t) \cup var(t')$, and for every set of formulas Φ, we define $var(\Phi) = \bigcup_{\varphi \in \Phi} var(\varphi)$.

Truth, Entailment. We fix $\mathbf{S} = \langle S, \wedge, 1, \leq \rangle$, a non-trivial meet-semilattice with one. By *Val*, we denote the set of *valuations*, i. e., the set of total functions from *Var* to S. We extend a valuation α to terms in the canonical way, i. e., $\alpha(\top) = 1$ and $\alpha(t_1 \sqcap t_2) = \alpha(t_1) \wedge \alpha(t_2)$. Note that we may view any term t as a total function from *Val* to S by defining the function application $t(\alpha)$ as $\alpha(t)$.

We say that a formula $t \sqsubseteq t'$ is *true* under a valuation α, denoted by $\alpha \models t \sqsubseteq t'$, iff $\alpha(t) \leq \alpha(t')$. We extend truth to sets of formulas, i. e., $\alpha \models \Phi$ iff $\alpha \models \varphi$ for all $\varphi \in \Phi$.

We say that Φ *entails* Ψ, denoted by $\Phi \models \Psi$, iff for all $\alpha \in Val$, $\alpha \models \Phi$ implies $\alpha \models \Psi$. We say that Φ is *equivalent* to Ψ, denoted by $\Phi \equiv \Psi$, iff $\Phi \models \Psi$ and $\Psi \models \Phi$. Note that in sets of formulas the operators top and meet are redundant on right-hand sides since for terms t, t_1', t_2', we have the equivalences $\{t \sqsubseteq \top\} \equiv \emptyset$ and $\{t \sqsubseteq t_1' \sqcap t_2'\} \equiv \{t \sqsubseteq t_1', t \sqsubseteq t_2'\}$.

Complexity of Entailment. We have defined entailment relative to a fixed semilattice $\mathbf{S}$. We may abstract from this structure and define entailment w. r. t. the variety of meet-semilattices with one. However, this would not change anything, as entailment w. r. t. that variety corresponds to entailment w. r. t. the two-element meet-semilattice $\mathbf{2} = \langle \{0, 1\}, \wedge, 1, \leq \rangle$, and vice versa. The deeper reason for this is a representation theorem [5, 20] which states that every meet-semilattice is isomorphic to a sub-meet-semilattice of a product of copies of $\mathbf{2}$. This implies that in every non-trivial meet-semilattice, uniform word problems yield the same answers as in $\mathbf{2}$. The same holds for entailment, for one can translate every formula $t \sqsubseteq t'$ into the equation $t \sqcap t' \doteq t$ and every equation $t \doteq t'$ into the set of formulas $\{t \sqsubseteq t', t' \sqsubseteq t\}$. As a consequence, deciding entailment is in Co-NP.

Actually given a finite set of formulas Φ and a formula ψ, deciding $\Phi \models \psi$ is linear in the size of Φ and ψ. To see this, note that we can flatten Φ and ψ such that all right-hand sides are variables[1] and every left-hand side is either a variable or a meet of two variables. This can be done by repeatedly replacing every subterm $x \sqcap y$ with $x, y \in Var$ by a new variable z and adding the equation $z \doteq x \sqcap y$ (translated into the three formulas $z \sqsubseteq x$, $z \sqsubseteq y$ and $x \sqcap y \sqsubseteq z$) to Φ. This transformation preserves entailment and the blow up of Φ is only linear in the size of the original formulas. Now, we may view $\mathbf{2}$ as the two-element boolean algebra because $\wedge$ corresponds to conjunction and (the characteristic function of) $\leq$ to implication. Thus in $\mathbf{2}$, the formula $x_1 \sqcap \ldots \sqcap x_m \sqsubseteq y_1 \sqcap \ldots \sqcap y_n$ may be seen as the propositional formula stating that the conjunction of the x_i implies the conjunction of the y_j. In

[1] We assume without loss of generality that $\Phi \cup \{\psi\}$ does not contain trivial formulas.

fact, if Φ and ψ are flattened, then each formula directly corresponds to a propositional Horn clause. Hence, the problem of deciding $\Phi \models \psi$ is reduced (with a linear blowup due to flattening) to entailment of propositional Horn clauses, which is known to be decidable in linear time [6].

Relations. Let X, Y and Z be sets and let $n \in \mathbb{N}$. Given n functions $f_1, \ldots, f_n : X \to Y$ and an n-ary function $g : Y^n \to Z$, we define the *n-ary composition* of g and $f_1, \ldots, f_n$ as the function $g[f_1, \ldots, f_n] : X \to Z$ such that for all $x \in X$, $g[f_1, \ldots, f_n](x) = g(f_1(x), \ldots, f_n(x))$.

Let $n \in \mathbb{N}$, let $t_1, \ldots, t_n$ be n terms and let $C : S^n \to \{0,1\}$, i.e., C is the characteristic function of an n-ary relation on S, the carrier of our fixed meet-semilattice $\mathbf{S}$. Viewing terms as functions from *Val* to S, the function $C[t_1, \ldots, t_n] : \mathit{Val} \to \{0,1\}$ is well-defined — it is the characteristic function of some set of valuations — and we say that C is *associated* with the terms $t_1, \ldots, t_n$. Note that for every enumeration $x_1, \ldots, x_m$ of a superset of the variables occuring in the terms $t_1, \ldots, t_n$ there is a unique function $C' : S^m \to \{0,1\}$ such that $C'[x_1, \ldots, x_m] = C[t_1, \ldots, t_n]$. Therefore, without loss of generality, we may assume that the associated terms are variables.

A relation Γ is an n-ary function $C : S^n \to \{0,1\}$ associated with n variables $x_1, \ldots, x_n$, i.e., $\Gamma = C[x_1, \ldots, x_n]$. We denote the set of variables occuring in Γ by $\mathit{var}(\Gamma)$, i.e., $\mathit{var}(\Gamma) = \{x_1, \ldots, x_n\}$.

We define a notion of truth for relations, similar to the one for formulas. We say that a relation $C[x_1, \ldots, x_n]$ is *true* under a valuation α, denoted by $\alpha \models C[x_1, \ldots, x_n]$, iff $C[x_1, \ldots, x_n](\alpha) = 1$. Rewriting this with the definition of n-ary composition, we see that $\alpha \models C[x_1, \ldots, x_n]$ iff $C(\alpha(x_1), \ldots, \alpha(x_n)) = 1$. We say that a relation Γ is *true* iff $\alpha \models \Gamma$ for all $\alpha \in \mathit{Val}$. Note that every set of formulas Φ may be expressed by an equivalent relation Γ_Φ with $\mathit{var}(\Gamma_\Phi) = \mathit{var}(\Phi)$, where for all valuations α, $\alpha \models \Phi$ iff $\alpha \models \Gamma_\Phi$. However, relations are strictly more expressive than formulas. For instance, inequality of two distinct variables x and y is not expressible by formulas, i.e., there is no set of formulas Φ such that for all $\alpha \in \mathit{Val}$, $\alpha \models \Phi$ iff $\alpha(x) \neq \alpha(y)$.

Inference Rules. An inference rule (or rule, for short) R is a triple $\langle \Phi, \psi, \Gamma \rangle$, where the *premises* Φ are a finite set of formulas, the *conclusion* ψ is a formula, and the *side condition* Γ is a relation. We say that R is *syntactic* iff the side condition Γ is true. We write a rule R as $R : \Phi/\psi$ if Γ or

$$R : \frac{\varphi_1 \quad \cdots \quad \varphi_m}{\psi} \text{ if } C[x_1, \ldots, x_n]$$

when $\Phi = \{\varphi_1, \ldots, \varphi_m\}$ and $\Gamma = C[x_1, \ldots, x_n]$. If R is syntactic then we may omit the side condition and write $R : \Phi/\psi$, simply. Without loss of generality we will assume that the right-hand sides of all premises are atomic and that

$var(\psi) \cup var(\Gamma) \subseteq var(\Phi)$, i.e., every variable of the conclusion or the side condition occurs in the premises.

Soundness, Completeness. Let $R : \Phi/\psi$ if Γ be an inference rule. We say that R is *sound* iff for all valuations α, $\alpha \models \Phi$ and $\alpha \models \Gamma$ implies $\alpha \models \psi$. We say that R is *syntactically sound* iff $\Phi \models \psi$. Note that syntactical soundness implies soundness, and every sound syntactic rule is syntactically sound. To point out a difference between the two notions of soundness, note that we can check algorithmically (and efficiently) whether a rule $R : \Phi/\psi$ if $C[x_1, \ldots, x_n]$ is syntactically sound, whereas checking whether R is sound may well be undecidable, depending on the properties of the meet-semilattice $\mathbf{S}$ and the function $C : S^n \to \{0, 1\}$.

We say that R is *complete* iff for all valuations α, $\alpha \models \Phi$ and $\alpha \models \psi$ implies $\alpha \models \Gamma$. Note that every syntactic rule is complete, trivially. Also note that for syntactically sound rules completeness is not an issue, as every syntactically sound rule $R : \Phi/\psi$ if Γ can be transformed by omitting the side condition into the (sound and complete) syntactic rule $R' : \Phi/\psi$. Hence, there is no reason why a syntactically sound rule should be restricted by a side condition.

Example 2. Assume that $\mathbf{S} = \langle S, \wedge, 1, \leq \rangle$ is the four-element meet-semilattice which is not a chain. Let s_1, s_2, p_1 and p_2 be four distinct variables, where we think of the s_i as representing systems and of the p_j as representing properties. We define the rules R_1 and R_2, where

$$R_k : \frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \text{ if } C_k[s_1, s_2, p_1, p_2]$$

and for all $a, b, c, d \in S$, $C_1(a, b, c, d) = 1$ iff c and d are incomparable, and $C_2(a, b, c, d) = 1$ iff $a \wedge b \leq c$ and $a \wedge b \leq d$. Both rules are sound as both side conditions are restrictive enough to prevent unsound circular reasoning. R_2 is trivially complete as $C_2[s_1, s_2, p_1, p_2]$ is equivalent to the conclusion. However, R_1 is incomplete as, for instance, it is not applicable to the (trivial) case when both properties equal 1. Note that the relation $C_1[s_1, s_2, p_1, p_2]$ is not expressible by formulas, which demonstrates that the language of side conditions is more expressive than languages of premises and conclusions. □

3 Assume-Guarantee Rules

Assume-Guarantee Rules. We call an inference rule $R : \Phi/\psi$ if Γ an assume-guarantee rule (or A-G rule, for short) iff for all premises $\varphi \in \Phi$, the left-hand side of ψ and the right-hand side of φ do not share any variables. We call an A-G rule $R : \Phi/\psi$ if Γ *circular* iff $\Phi \not\models \psi$.

$$R_3 : \frac{s_1 \sqsubseteq p_1 \qquad s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \qquad R_6 : \frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2}$$

$$R_4 : \frac{s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \qquad R_7 : \frac{p_3 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2}$$

$$R_5 : \frac{p_3 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{p_3 \sqcap s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \qquad R_8 : \frac{p_1 \sqcap s_1 \sqsubseteq \top \qquad p_2 \sqsubseteq s_2}{p_1 \sqcap s_1 \sqsubseteq s_2 \sqcap p_2}$$

Figure 1: Sample assume-guarantee rules

Example 3. As the definition of A-G rules does not involve the side condition, we may illustrate it using syntactic rules only, see figure 1. There, s_1, s_2, p_1, p_2 and p_3 are five distinct variables, where the system/property distinction is as in example 2.

The rules R_3, R_4 and R_5 are non-circular A-G rules. Note the second premise of R_4, which may be read as *assuming the property* p_1 *the system* s_2 *guarantees the property* p_2. Likewise, the conclusion of R_5 may be read as *assuming* p_3 *the composition of* s_1 *and* s_2 *guarantees both* p_1 *and* p_2. This should explain where the term *assume-guarantee rule* comes from.

The rules R_6, R_7 and R_8 (and also R_1 and R_2 from example 2) are circular A-G rules as they are not syntactically sound. The term *circular* is justified for R_6, whose premises express circular assume-guarantee dependencies between the properties p_1 and p_2. For R_7 and R_8, however, there is no circularity in the premises. In the case of R_7, unsoundness arises from the unresolved assumption p_3; compare to R_5 where that assumption is resolved. R_8 is unsound because it is nonsense, it serves to demonstrate that not every assume-guarantee rule has a meaningful reading. So, the term *circular* should not be taken literally, rather it is an abstraction capturing the most important property of circular assume-guarantee reasoning, namely its lack of syntactical soundness. □

The following propositions provide an alternative characterization of circularity resp. a sufficient criterion for the truth of the premises of an A-G rule.

Proposition 1. *An A-G rule* $R : \Phi/t_\psi \sqsubseteq t_\psi{}'$ if Γ *is circular if and only if* $\Phi \not\models t_\psi \sqsubseteq x$ *for some* $x \in var(t_\psi{}')$.

Proof. This follows from the obvious converse that $\Phi \models t_\psi \sqsubseteq t_\psi{}'$ if and only if for all $x \in var(t_\psi{}')$, $\Phi \models t_\psi \sqsubseteq x$. □

Proposition 2. *Let* $R : \Phi/t_\psi \sqsubseteq t_\psi{}'$ if Γ *be an A-G rule and let* α *be a valuation. If for all* $x, y \in var(\Phi) \setminus var(t_\psi)$, *1.* $\Phi \models t_\psi \sqsubseteq x$ *implies* $\alpha(x) = 1$*, and*

2. $\Phi \not\models t_\psi \sqsubseteq x$ *and* $\Phi \not\models t_\psi \sqsubseteq y$ *implies* $\alpha(x) = \alpha(y)$*,*

then $\alpha \models \Phi$*.*

Proof. Assume that the conditions 1 and 2 hold for all $x, y \in var(\Phi) \setminus var(t_\psi)$. Let $t \sqsubseteq t' \in \Phi$. If $t' = \top$ then obviously $\alpha \models t \sqsubseteq t'$. If $t' \neq \top$ then $t' = x$ for some $x \in var(\Phi)$ because we assume all right-hand sides of premises to be atomic. Furthermore, $x \notin var(t_\psi)$ by definition of A-G rules. Case distinction.

- $\Phi \models t_\psi \sqsubseteq x$. Then $\alpha(x) = 1$ by condition 1, so $\alpha(t) \leq \alpha(t')$, i.e., $\alpha \models t \sqsubseteq t'$.
- $\Phi \not\models t_\psi \sqsubseteq x$. Then there exists $y \in var(t)$ such that $\Phi \not\models t_\psi \sqsubseteq y$. To see this, assume that $\Phi \models t_\psi \sqsubseteq y$ for all $y \in var(t)$; this would mean $\Phi \models t_\psi \sqsubseteq t$, which by $t \sqsubseteq x \in \Phi$ would yield the contradiction $\Phi \models t_\psi \sqsubseteq x$. Furthermore, $y \notin var(t_\psi)$ as otherwise $\Phi \models t_\psi \sqsubseteq y$ trivially. Hence, we know that $\alpha(x) = \alpha(y)$ by condition 2, which yields $\alpha(t) \leq \alpha(t')$, i.e., $\alpha \models t \sqsubseteq t'$. □

A-G Rules for Compositional Verification. As already hinted in example 3, for the purpose of verification we distinguish systems and properties, so we partition our variable set *Var* into *system variables* s_i and *property variables* p_j. Section 4 will show that already the composition of only two systems exhibits the incompleteness of compositional circular assume-guarantee reasoning, so actually we can restrict the variable set to $Var = \{s_1, s_2\} \uplus \{p_1, p_2, p_3, \dots\}$.

The goal of compositional verification is to establish that the composition of some systems (in our case, s_1 and s_2) guarantees some property (possibly assuming some other property). So for an A-G rule to be useful for compositional verification, $s_1 \sqcap s_2$ must be a subterm of the left-hand side of the conclusion, which we will implicitly assume henceforth. Thus, without loss of generality we may assume that an A-G rule R is presented in the form

$$R : \frac{\varphi_1 \quad \dots \quad \varphi_m}{t_\psi \sqsubseteq t_\psi{}'} \text{ if } C[s_1, s_2, p_1, \dots, p_n]$$

where $s_1 \sqcap s_2$ occurs as a subterm in t_ψ and $var(\{\varphi_1, \dots, \varphi_m, t_\psi \sqsubseteq t_\psi{}'\}) = \{s_1, s_2, p_1, \dots, p_n\}$. The latter requirement can always be achieved by renaming some property variables and extending and reordering the associated variables in the side condition. By the definition of A-G rules, $var(t_\psi) \cap var(t') = \emptyset$ for every premise $t \sqsubseteq t'$, so $t' \in \{\top, p_1, \dots, p_n\}$ as we assume the right-hand sides of premises to be atomic.

Compositionality. Let $R : \Phi/\psi$ if $C[s_1, s_2, p_1, \ldots, p_n]$ be an A-G rule. We will call R compositional if it avoids the system composition $s_1 \sqcap s_2$ in the premises as well as in the side condition. Formally, we say that R is *compositional in the premises* iff $s_1 \sqcap s_2$ is not a subterm of any left-hand side in Φ. We say that R is *compositional in the side condition* iff $C[s_1, s_2, p_1, \ldots, p_n]$ is expressible as a boolean combination of relations whose associated variables either do not include s_1 or s_2. I. e., R is compositional in the side condition iff there are $r_1, r_2 \in \mathbb{N}$, a (r_1+r_2)-ary boolean function $F : \{0,1\}^{r_1+r_2} \to \{0,1\}$ and $r_1 + r_2$ $(n+1)$-ary functions $C_1^1, \ldots, C_1^{r_1}, C_2^1, \ldots, C_2^{r_2} : S^{n+1} \to \{0,1\}$ such that

$$C[s_1, s_2, \tilde{p}] = F\left[C_1^1[s_1, \tilde{p}], \ldots, C_1^{r_1}[s_1, \tilde{p}], C_2^1[s_2, \tilde{p}], \ldots, C_2^{r_2}[s_2, \tilde{p}]\right] \qquad (3)$$

where $\tilde{p}$ abbreviates the enumeration $p_1, \ldots, p_n$. We say that R is *compositional* iff it is compositional in the premises and in the side condition.

One may think of the above functions C_i^k as abstracting the system s_i together with the properties $p_1, \ldots, p_n$ to a boolean value. Actually, we can relax the above definition of compositionality in the side condition from boolean to arbitrary finitary abstractions C_i^k. I. e., R is compositional in the side condition iff there are a finite set D and $r_1, r_2 \in \mathbb{N}$ and $F : D^{r_1+r_2} \to \{0,1\}$ and $C_1^1, \ldots, C_1^{r_1}, C_2^1, \ldots, C_2^{r_2} : S^{n+1} \to D$ such that the equation (3) holds.

Example 4. Recall the A-G rules R_1 to R_8 from the examples 2 and 3. All these rules are compositional in the premises, and the syntactic rules R_3 to R_8 are compositional in the side condition, trivially. The rule R_1 is also compositional in the side condition but R_2 is not. □

4 Incompleteness of Compositional Rules

Before we go about to establish our main incompleteness result, we need to set up two technical pieces of machinery. First, we prove Lemma 3, a purely combinatorial proposition about the infeasibility of certain boolean equation systems. Second, we introduce special substructures of a meet-semilattice, the so-called forks.

Lemma 3. *Let $m, n \in \mathbb{N}$ and $F : \{0,1\}^{m+n} \to \{0,1\}$. Then the system of equations E_F over the variables u_i^k $(0 \le i \le 2^{\min\{m,n\}}, 1 \le k \le m)$ and v_j^l $(0 \le j \le 2^{\min\{m,n\}}, 1 \le l \le n)$ has no solutions, where E_F is defined as*

$$\begin{aligned} E_F = \{F(u_i^1, \ldots, u_i^m, v_j^1, \ldots, v_j^n) = 1 \mid 0 \le i, j \le 2^{\min\{m,n\}}, i \ne j\} \\ \cup \{F(u_i^1, \ldots, u_i^m, v_i^1, \ldots, v_i^n) = 0 \mid 1 \le i \le 2^{\min\{m,n\}}\}. \end{aligned}$$

Proof. Obviously, the system E_F has solutions if and only if the system

$$\begin{aligned} \{F'(u_i^1, \ldots, u_i^n, v_j^1, \ldots, v_j^m) = 1 \mid 0 \le i, j \le 2^{\min\{m,n\}}, i \ne j\} \\ \cup \{F'(u_i^1, \ldots, u_i^n, v_i^1, \ldots, v_i^m) = 0 \mid 1 \le i \le 2^{\min\{m,n\}}\} \end{aligned}$$

over the variables u_i^l and v_j^k $(0 \le i, j \le 2^{\min\{m,n\}}, 1 \le l \le n, 1 \le k \le m)$ has solutions, where $F' : \{0,1\}^{m+n} \to \{0,1\}$ is defined by swapping the first m and the last n arguments to F, i.e., $F'(b^1, \dots, b^n, a^1, \dots, a^m) = F(a^1, \dots, a^m, b^1, \dots, b^n)$ for all $a^1, \dots, a^m, b^1, \dots, b^n \in \{0,1\}$. Therefore, we can without loss of generality assume that $m = \min\{m, n\}$ and proceed by induction on m.

- $m = 0$. Then $2^m = 1$ so E_F contains the equations $F(v_1^1, \dots, v_1^n) = 1$ and $F(v_1^1, \dots, v_1^n) = 0$, hence E_F cannot have solutions.
- $m > 0$. Towards a contradiction assume that there exists a function $F : \{0,1\}^{m+n} \to \{0,1\}$ such that E_F has solutions. Fix such a solution and let $a_i^m \in \{0,1\}$ be the value of u_i^m $(0 \le i \le 2^m)$. Then obviously, the system

$$\begin{aligned} &\{F(u_i^1, \dots, u_i^{m-1}, a_i^m, v_j^1, \dots, v_j^n) = 1 \mid 0 \le i, j \le 2^m, i \ne j\} \\ \cup\, &\{F(u_i^1, \dots, u_i^{m-1}, a_i^m, v_i^1, \dots, v_i^n) = 0 \mid 1 \le i \le 2^m\} \end{aligned} \tag{4}$$

also has solutions. Let $I_0, I_1 \subseteq \{0, \dots, 2^m\}$ such that $I_0 = \{i \mid a_i^m = 0\}$ and $I_1 = \{i \mid a_i^m = 1\}$. As $|I_0| + |I_1| = 2^m + 1$, there is $a \in \{0,1\}$ such that $|I_a| \ge 2^{m-1} + 1$; let $I'_a \subseteq I_a$ such that $|I'_a| = 2^{m-1} + 1$. Because the system (4) has solutions, the subsystem

$$\begin{aligned} &\{F(u_i^1, \dots, u_i^{m-1}, a, v_j^1, \dots, v_j^n) = 1 \mid i, j \in I'_a, i \ne j\} \\ \cup\, &\{F(u_i^1, \dots, u_i^{m-1}, a, v_i^1, \dots, v_i^n) = 0 \mid i \in I'_a, i \ne \min I'_a\} \end{aligned} \tag{5}$$

also has solutions. Let $\pi : \{0, \dots, 2^{m-1}\} \to I'_a$ be a bijection with $\pi(0) = \min I'_a$, and define $F_a : \{0,1\}^{m+n-1} \to \{0,1\}$ by

$$F_a(a^1, \dots, a^{m-1}, b^1, \dots, b^n) = F(a^1, \dots, a^{m-1}, a, b^1, \dots, b^n)$$

for all $a^1, \dots, a^{m-1}, b^1, \dots, b^n \in \{0,1\}$. Thus, (5) can be written as

$$\begin{aligned} &\{F_a(u_{\pi(i)}^1, \dots, u_{\pi(i)}^{m-1}, v_{\pi(j)}^1, \dots, v_{\pi(j)}^n) = 1 \mid 0 \le i, j \le 2^{m-1}, i \ne j\} \\ \cup\, &\{F_a(u_{\pi(i)}^1, \dots, u_{\pi(i)}^{m-1}, v_{\pi(i)}^1, \dots, v_{\pi(i)}^n) = 0 \mid 1 \le i \le 2^{m-1}\}. \end{aligned} \tag{6}$$

We reach a contradiction, as the system of equations E_{F_a}, which equals (6) up to variable renaming, does not have solutions by induction hypothesis. □

Forks. We say that $Y \subseteq S$ is a *fork* iff there is $x \in Y$ such that for all $y, z \in Y$, $y \ne z$ implies $x = y \wedge z$; if Y is infinite then we say that Y is a fork of *infinite width*, otherwise the size of Y is $m \in \mathbb{N}$ and we say that Y is a fork of *width* $m - 1$. Note that if $\mathbf{S}$ contains a fork of infinite width then it also contains forks of width m for every $m \in \mathbb{N}$.

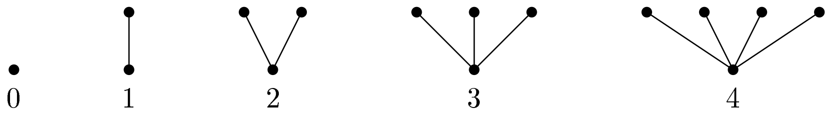

Figure 2: Forks of width 0 to 4

Example 5. Some forks of finite width are depicted in figure 2. Note that if **S** is a chain then it contains only forks of width 1, and if **S** is the power set meet-semilattice of an arbitrary set X then it contains forks of infinite width iff X is infinite. In particular, the meet-semilattice of ω-languages over some alphabet Σ (see example 1) contains forks of infinite width iff Σ is not unary. The same holds for the meet-semilattice of prefix-closed $*$-languages over Σ. This is so because if Σ is unary then the prefix-closed $*$-languages form a chain. And if Σ contains the distinct letters a and b, then $Y = \{a^*\} \cup \{a^* \cup a^i b^* \mid i \in \mathbb{N}\}$ is a fork of infinite width. $\square$

Incompleteness Lemma. Our proof of the incompleteness of sound and compositional circular A-G rules works by contradiction. Its main part (Lemma 4) reduces the existence of a sound and complete circular A-G rule R which is compositional in the side condition to the feasibility of a boolean equation system, which is known to be be infeasible by Lemma 3. The resulting contradiction implies that R must be unsound or incomplete.

For the reduction to work, Lemma 4 has to assume that the meet-semilattice **S** contains forks of sufficient width. However, this is not a restriction as all interesting semilattices (in verification) in fact do contain forks of infinite width, see example 5.

Lemma 4. *Let $R : \Phi/\psi$ if $C[s_1, s_2, p_1, \ldots, p_n]$ be a circular A-G rule. Let $r_1, r_2 \in \mathbb{N}$, let $F : \{0,1\}^{r_1+r_2} \to \{0,1\}$ be a $(r_1 + r_2)$-ary boolean function and let $C_1^1, \ldots, C_1^{r_1}, C_2^1, \ldots, C_2^{r_2} : S^{n+1} \to \{0,1\}$ be $(n+1)$-ary functions such that*

$$C[s_1, s_2, \tilde{p}] = F\left[C_1^1[s_1, \tilde{p}], \ldots, C_1^{r_1}[s_1, \tilde{p}], C_2^1[s_2, \tilde{p}], \ldots, C_2^{r_2}[s_2, \tilde{p}]\right] \qquad (7)$$

*where $\tilde{p}$ stands for $p_1, \ldots, p_n$. If **S** contains a fork of width $2^{\min\{r_1,r_2\}}$ then R is unsound or incomplete.*

Proof. Let ψ be $t_\psi \sqsubseteq t_\psi{}'$ and recall that $var(\Phi \cup \{\psi\}) = \{s_1, s_2, p_1, \ldots, p_n\}$. Without loss of generality we may assume that there is some $m \in \{0, \ldots, n\}$ such that for all $j \geq 1$, $\Phi \models t_\psi \sqsubseteq p_j$ iff $j \leq m$; this can always be achieved by renaming some property variables.

We will prove this lemma by contradiction, so assume that R is sound and complete and let $Y \subseteq S$ be a fork of width $2^{\min\{r_1,r_2\}}$, i.e., $Y =$

$\{x_0, x_1, \ldots, x_{2^r}\}$ with $r = \min\{r_1, r_2\}$ such that for all $i, j \geq 0$ with $i \neq j$, $x_0 = x_i \wedge x_j$. By Lemma 3, we know that the system of equations

$$\begin{aligned} &\{F(u_i^1, \ldots, u_i^{r_1}, v_j^1, \ldots, v_j^{r_2}) = 1 \mid 0 \leq i, j \leq 2^{\min\{r_1, r_2\}}, i \neq j\} \\ \cup\, &\{F(u_i^1, \ldots, u_i^{r_1}, v_i^1, \ldots, v_i^{r_2}) = 0 \mid 1 \leq i \leq 2^{\min\{r_1, r_2\}}\} \end{aligned} \tag{8}$$

over the variables u_i^k and v_j^l ($0 \leq i, j \leq 2^{\min\{r_1, r_2\}}, 1 \leq k \leq r_1, 1 \leq l \leq r_2$) has no solutions. However, we will show that there is a solution to (8), namely with $C_1^k(x_i, \widetilde{1}, \widetilde{x_0})$ resp. $C_2^l(x_j, \widetilde{1}, \widetilde{x_0})$ as the values of u_i^k resp. v_j^l, where $\widetilde{1}$ abbreviates the enumeration $1, \ldots, 1$ of length m and $\widetilde{x_0}$ abbreviates the enumeration $x_0, \ldots, x_0$ of length $n - m$. To see that this is a solution, for every $i, j \in \{0, \ldots, 2^{\min\{r_1, r_2\}}\}$ we choose a valuation α_{ij} such that $\alpha_{ij}(s_1) = x_i$, $\alpha_{ij}(s_2) = x_j$, $\alpha_{ij}(p_1) = \cdots = \alpha_{ij}(p_m) = 1$ and $\alpha_{ij}(p_{m+1}) = \cdots = \alpha_{ij}(p_n) = x_0$.

First, let $i, j \in \{0, \ldots, 2^{\min\{r_1, r_2\}}\}$ with $i \neq j$. Because of the choice of α_{ij}, Proposition 2 yields that $\alpha_{ij} \models \Phi$. As $s_1 \sqcap s_2$ is a subterm of t_ψ, we have $\alpha_{ij}(t_\psi) \leq \alpha_{ij}(s_1 \sqcap s_2) = x_i \wedge x_j = x_0$. And as $x_0 \leq \alpha_{ij}(t)$ for every term t, we have $\alpha_{ij}(t_\psi) \leq \alpha_{ij}(t_\psi')$, i.e., $\alpha_{ij} \models \psi$. Hence, premises and conclusion of R are true under α_{ij}. Due to completeness, the side condition must be true as well, so $\alpha_{ij} \models C[s_1, s_2, \tilde{p}]$, i.e., $C[s_1, s_2, \tilde{p}](\alpha_{ij}) = 1$. With (7), the decomposition of the side condition $C[s_1, s_2, \tilde{p}]$, this yields the equation

$$F\big(C_1^1(x_i, \widetilde{1}, \widetilde{x_0}), \ldots, C_1^{r_1}(x_i, \widetilde{1}, \widetilde{x_0}), C_2^1(x_j, \widetilde{1}, \widetilde{x_0}), \ldots, C_2^{r_2}(x_j, \widetilde{1}, \widetilde{x_0})\big) = 1.$$

Second, let $i \in \{1, \ldots, 2^{\min\{r_1, r_2\}}\}$. Again, Proposition 2 yields that $\alpha_{ii} \models \Phi$. For every $p_j \in \mathit{var}(t_\psi)$, $\Phi \models t_\psi \sqsubseteq p_j$ trivially, so $\alpha_{ii}(p_j) = 1$. Therefore, $\alpha_{ii}(t_\psi) = \alpha_{ii}(s_1 \sqcap s_2) = x_i \wedge x_i = x_i$. On the other hand by Proposition 1, there is some $p_j \in \mathit{var}(t_\psi')$ such that $\Phi \not\models t_\psi \sqsubseteq p_j$, so $\alpha_{ii}(p_j) = x_0$, which implies $\alpha_{ii}(t_\psi') \leq x_0$. From the definition of forks, we know that $x_0 \leq x_i$ and $x_0 \neq x_i$, so $x_i \not\leq x_0$. Therefore, $\alpha_{ii}(t_\psi) \not\leq \alpha_{ii}(t_\psi')$, i.e., $\alpha_{ii} \not\models \psi$. Hence, the premises of R are true under α_{ii} but the conclusion is not. Due to soundness, the side condition must not be true, so $\alpha_{ii} \not\models C[s_1, s_2, \tilde{p}]$, i.e., $C[s_1, s_2, \tilde{p}](\alpha_{ii}) = 0$. Using (7) again, we get the equation

$$F\big(C_1^1(x_i, \widetilde{1}, \widetilde{x_0}), \ldots, C_1^{r_1}(x_i, \widetilde{1}, \widetilde{x_0}), C_2^1(x_i, \widetilde{1}, \widetilde{x_0}), \ldots, C_2^{r_2}(x_i, \widetilde{1}, \widetilde{x_0})\big) = 0.$$

Thus, the system of equations (8) indeed has a solution, which ends this proof by contradiction. □

We have shown that an A-G rule which is compositional in the side condition cannot be both sound and complete — provided that the semilattice **S** contains forks of sufficient width. The latter is the case trivially whenever **S** contains a fork of infinite width.

Theorem 5. *If* **S** *contains forks of infinite width then there exists no sound and complete compositional circular assume-guarantee rule.*

Proof. Follows from Lemma 4. □

5 Discussion of Related Work

Incompleteness of Other Rules. Our setting of inference rules in meet-semilattices is a very abstract one. Most circular A-G rules in the literature are presented in more concrete settings, i. e., they use more structure than just meet and order — and that extra structure is usually indispensable for proving soundness by a circularity-breaking induction. This raises the question to what extent our incompleteness result is relevant for such concrete rules.

We claim that most circular A-G rules can be transformed — preserving soundness and compositionality — into equivalent circular A-G rules in meet-semilattices which contain forks of infinite width. Incompleteness of the transformed rule then points out a defect of the original rule: There must be cases in which the original rule is not applicable although soundness is not in danger. Below, we will exemplify two such transformations.

Various circular A-G rules have been proposed for settings, where systems and properties are presented as some form of transition graphs enriched with input and output, e. g., Moore or Mealy machines [13, 11] or Reactive Modules [4]. These rules establish certain refinement relations, e. g., trace containment or simulation, between compositions of transition graphs. Thereby, composition is a partial operation, which is defined only if the components satisfy some condition called compatibility. These compatibilities form an implicit side condition to the A-G rules, which is made explicit by the transformation. We demonstrate this in the following example by means of the transformation of a circular A-G rule for Moore machines.

Example 6. Let $\mathcal{X}$ be a finite set of variables, ranging over an arbitrary non-empty domain $\mathcal{D}$. A *Moore machine* M is a (possibly infinite) state transition graph with input variables $I_M \subseteq \mathcal{X}$ and output variables $O_M \subseteq \mathcal{X}$, where the nodes resp. edges of the graph are labeled by valuations of the output resp. input variables; for a formal definition see for instance [11, 13]. Naturally, one associates a trace language $[\![M]\!] \subseteq \Sigma^*$ with M, where $\Sigma = \mathcal{D}^{\mathcal{X}}$ is the set of valuations of all variables. The parallel composition $M_1 \parallel M_2$ of two Moore machines M_1 and M_2 corresponds to language intersection, i. e., $[\![M_1 \parallel M_2]\!] = [\![M_1]\!] \cap [\![M_2]\!]$. Note that $M_1 \parallel M_2$ is defined if and only if M_1 and M_2 are compatible, i. e., O_{M_1} and O_{M_2} are disjoint.

For Moore machines with trace semantics, the following circular proof rule is known:

$$\frac{[\![P_2 \parallel S_1]\!] \subseteq [\![P_1]\!] \qquad [\![P_1 \parallel S_2]\!] \subseteq [\![P_2]\!]}{[\![S_1 \parallel S_2]\!] \subseteq [\![P_1 \parallel P_2]\!]} \tag{9}$$

where S_1, S_2, P_1, P_2 are Moore machines such that all parallel compositions in (9) are defined. We transform this rule into the A-G rule R_{Moore} for the meet-semilattice $\langle \mathcal{P}(\Sigma^*), \cap, \Sigma^*, \subseteq \rangle$, the power set of Σ^*:

$$R_{\text{Moore}} : \frac{p_2 \sqcap s_1 \sqsubseteq p_1 \qquad p_1 \sqcap s_2 \sqsubseteq p_2}{s_1 \sqcap s_2 \sqsubseteq p_1 \sqcap p_2} \text{ if } F\big[C[s_1], C[p_1], C[s_2], C[p_2]\big]$$

where $D = \mathcal{P}(\mathcal{X}) \uplus \{\bot\}$ is a finite set and $C : \mathcal{P}(\Sigma^*) \to D$ is a finitary abstraction mapping each $L \in \mathcal{P}(\Sigma^*)$ to the least set of output variables O_M such that M is a Moore machine with $[\![M]\!] = L$; if no such Moore machine exists then $C(L) = \bot$. The function $F : D^4 \to \{0, 1\}$ is defined by $F(O_{s_1}, O_{p_1}, O_{s_2}, O_{p_2}) = 1$ iff

$$\begin{gathered} O_{s_1} \neq \bot \text{ and } O_{p_1} \neq \bot \text{ and } O_{s_2} \neq \bot \text{ and } O_{p_2} \neq \bot \\ \text{and } O_{s_1} \cap O_{s_2} = O_{p_1} \cap O_{p_2} = O_{p_2} \cap O_{s_1} = O_{p_1} \cap O_{s_2} = \emptyset. \end{gathered}$$

R_{Moore} is a compositional circular A-G rule according to Section 3. Circularity and compositionality in the premises are obvious. Compositionality in the side condition holds as obviously there exist functions $C_i^k : \mathcal{P}(\Sigma^*)^3 \to D$ such that

$$\begin{aligned} &F\big[C[s_1], C[p_1], C[s_2], C[p_2]\big] \\ &\qquad = F\big[C_1^1[s_1, p_1, p_2], C_1^2[s_1, p_1, p_2], C_2^1[s_2, p_1, p_2], C_2^2[s_2, p_1, p_2]\big]. \end{aligned}$$

Moreover, soundness of R_{Moore} can be reduced to soundness of the original rule (9), and vice versa, so both rules are applicable in exactly the same cases. To see how soundness of R_{Moore} reduces to (9), consider the premises and side condition of R_{Moore} to be true under a valuation α. Then there are Moore machines S_i and P_j such that $[\![S_i]\!] = \alpha(s_i)$ and $[\![P_j]\!] = \alpha(p_j)$ and S_1 and S_2, P_1 and P_2, P_2 and S_1 as well as P_1 and S_2 are compatible, i. e., all parallel compositions in (9) are defined. Furthermore, we have $[\![P_2]\!] \cap [\![S_1]\!] \subseteq [\![P_1]\!]$ and $[\![P_1]\!] \cap [\![S_2]\!] \subseteq [\![P_2]\!]$, which by language intersection and soundness of (9) implies $[\![S_1 \parallel S_2]\!] \subseteq [\![P_1 \parallel P_2]\!]$, which in turn by language intersection implies that the conclusion of R_{Moore} is true under α. To show the converse reduction, let S_i and P_j be Moore machines such that all parallel compositions in (9) are defined, i. e., S_1 and S_2, P_1 and P_2, P_2 and S_1 as well as P_1 and S_2 are compatible. Obviously, soundness of (9) follows by language intersection and soundness of R_{Moore}.

As the proof rule (9) has been proven sound in [13], R_{Moore} is a sound and compositional circular A-G rule. Thus by Theorem 5, R_{Moore} is incomplete because the meet-semilattice $\langle \mathcal{P}(\Sigma^*), \cap, \Sigma^*, \subseteq \rangle$ contains forks of infinite width. Hence there are cases in which circular reasoning is admissible yet the rule (9) is not applicable, due to partiality of parallel composition. □

Other kinds of circular A-G rules focus on temporal logics to present properties (and sometimes systems also), see for instance [1, 2, 12]. In order to break the circularity, such rules usually employ so-called assume-guarantee specifications, i. e., formulas of the form $\varphi \triangleright \psi$ where $\triangleright$ is a special temporal operator ensuring that during any computation the guarantee ψ holds at least one step longer than the assumption φ. In our meet-semilattice setting, we cannot express A-G specifications in the premises of inference rules.

However, we can move A-G specifications to the side condition, where their truth is expressible as a relation. In the following example, we demonstrate this transformation on a simple circular rule for A-G specifications.

Example 7. Let AP be a non-empty set of atomic propositions. We say that $\Sigma = \mathcal{P}(AP)$ is the set of states, and Σ^ω is the set of computations. A system is a set of computations, and the parallel composition of two systems S_1 and S_2 is their intersection $S_1 \cap S_2$. Likewise, a property is a set of computations, and we say that a property P entails another property Q iff $P \subseteq Q$. We may represent certain properties by formulas in linear-time temporal logic (LTL), which are constructed from atomic propositions by means of boolean operators and the standard temporal operators X (next-time), U (until), F (eventually) and G (always); for a formal definition of syntax and semantics of LTL see for instance [7]. Henceforth, we will identify a formula φ with the property it represents.

Given two formulas φ and ψ, we define the *assume-guarantee specification* $\varphi \rhd \psi$ as an abbreviation of the formula $\neg(\varphi \mathbin{\mathsf{U}} \neg\psi)$, cf. [19]. The temporal operator $\rhd$ satisfies the following equalities and inequalities:

$$\varphi \rhd \psi = \psi \wedge \big(\varphi \Rightarrow \mathsf{X}(\varphi \rhd \psi)\big) \tag{10}$$

$$\mathsf{G}\varphi \wedge (\varphi \rhd \psi) \subseteq \mathsf{G}\psi \tag{11}$$

From the fix-point equation (10), we can read off that $\varphi \rhd \psi$ is the weakest property where ψ holds strictly longer than φ along every computation.

For A-G specifications, the following circular proof rule is known:

$$\frac{S_1 \subseteq \varphi_2 \rhd \varphi_1 \qquad S_2 \subseteq \varphi_1 \rhd \varphi_2}{S_1 \cap S_2 \subseteq \mathsf{G}(\varphi_1 \wedge \varphi_2)} \tag{12}$$

where S_1, S_2 are systems and φ_1, φ_2 are LTL formulas. We transform this rule into the A-G rule $R_\rhd$ for the meet-semilattice of systems and properties $\langle \mathcal{P}(\Sigma^\omega), \cap, \Sigma^\omega, \subseteq \rangle$:

$$R_\rhd : \frac{p_4 \sqcap s_1 \sqsubseteq p_3 \qquad p_3 \sqcap s_2 \sqsubseteq p_4}{s_1 \sqcap s_2 \sqsubseteq p_3 \sqcap p_4} \text{ if } C_1^1[s_1, p_1, \ldots, p_4] * C_2^1[s_2, p_1, \ldots, p_4]$$

where $* : \{0,1\}^2 \to \{0,1\}$ denotes multiplication (i. e., conjunction in logical terms) and the functions $C_i^k : \mathcal{P}(\Sigma^\omega)^5 \to \{0,1\}$ are defined by

$$C_1^1(S_1, P_1, P_2, P_3, P_4) = 1 \quad \text{iff} \quad P_3 = \mathsf{G}P_1 \text{ and } P_4 = \mathsf{G}P_2 \text{ and } S_1 \subseteq P_2 \rhd P_1,$$
$$C_2^1(S_2, P_1, P_2, P_3, P_4) = 1 \quad \text{iff} \quad P_3 = \mathsf{G}P_1 \text{ and } P_4 = \mathsf{G}P_2 \text{ and } S_2 \subseteq P_1 \rhd P_2.$$

Note that the equality $P_3 = \mathsf{G}P_1$ is supposed to hold iff there exists an LTL formula φ_1 such that φ_1 and $\mathsf{G}\varphi_1$ represent the properties P_1 and P_3, respectively; $P_4 = \mathsf{G}P_2$ is to be interpreted similarly.

Obviously, $R_{\rhd}$ is a compositional circular A-G rule[2] according to Section 3. Moreover, soundness of $R_{\rhd}$ can be reduced to soundness of the original rule (12), and vice versa, so both rules are applicable in exactly the same cases. To see how soundness of $R_{\rhd}$ reduces to (12), consider the premises and side condition of $R_{\rhd}$ to be true under a valuation α. Then there are LTL formulas φ_j such that $\mathsf{G}\varphi_1 = \alpha(p_3)$ and $\mathsf{G}\varphi_2 = \alpha(p_4)$ and $\alpha(s_1) \subseteq \varphi_2 \rhd \varphi_1$ and $\alpha(s_2) \subseteq \varphi_1 \rhd \varphi_2$. Using soundness of (12), we infer $\alpha(s_1) \cap \alpha(s_2) \subseteq \mathsf{G}(\varphi_1 \wedge \varphi_2)$, which implies that the conclusion of $R_{\rhd}$ is true under α. To show the converse reduction, let S_i and φ_j be systems and formulas, respectively, such that the premises of rule (12) hold. By (11), these premises imply $\mathsf{G}\varphi_2 \cap S_1 \subseteq \mathsf{G}\varphi_1$ and $\mathsf{G}\varphi_1 \cap S_2 \subseteq \mathsf{G}\varphi_2$, respectively. Using soundness of $R_{\rhd}$, we infer $S_1 \cap S_2 \subseteq \mathsf{G}\varphi_1 \cap \mathsf{G}\varphi_2$, which is equivalent to the conclusion of (12).

As the proof rule (12) is sound, cf. [17, 19], $R_{\rhd}$ is a sound and compositional circular A-G rule. Thus by Theorem 5, $R_{\rhd}$ is incomplete because the meet-semilattice $\langle \mathcal{P}(\Sigma^\omega), \cap, \Sigma^\omega, \subseteq \rangle$ contains forks of infinite width. As a consequence, the rule (12) does not capture all sound circular reasoning patterns, i. e., there are cases in which circular reasoning is admissible yet (12) is not applicable. $\square$

In short, this paper shows that compositionality implies incompleteness. Yet, we did not encounter any complete rule except for the rather trivial rule R_2 from example 2. This raises the question whether non-trivial sound and complete circular A-G rules do exist at all. They do — in [14], we present a very general sound and complete circular A-G rule for certain classes of lattices. Of course, that rule must be non-compositional; in fact, it is non-compositional both in the premises and in the side condition. Still, that general rule can be instantiated to many known circular A-G rules, no matter whether they are compositional or not.

Other Notions of Completeness. When some complex system should be verified against a conjunction of properties, one usually applies backward reasoning, i. e., one matches the verification goal against the conclusion of a proof rule and from the premises and the side condition one infers the subgoals that need to be established. In [19], the authors investigate a notion of completeness that characterizes rules which always enable backward reasoning, so we will term this notion backward completeness. Adopted to our setting, a rule $R : \Phi/\psi$ if Γ is called *backward complete* iff for all valuations α, $\alpha \models \psi$ implies $\alpha' \models \Phi$ and $\alpha' \models \Gamma$ for some valuation α' which agrees with α on the variables of ψ. Thus, truth of the conclusion implies that the premises and the side condition can be made true through choosing (i. e., guessing) appropriate values for the *auxiliary variables*, i. e., for those

[2] The trivial premises $p_1 \sqsubseteq \top$ and $p_2 \sqsubseteq \top$ have been omitted from the definition of $R_{\rhd}$ for the sake of readability.

variables in the premises that do not occur in the conclusion. Note that backward completeness does not distinguish premises and side condition, whereas this distinction is essential for our notion of completeness.

Our notion of completeness relates more to forward reasoning, i. e., from prior knowledge which subsystems guarantee which properties assuming which other properties, we want to infer that the complex system guarantees a conjunction of properties. A complete rule (in the sense of this paper) will enable this inference whenever the conclusion is consistent with our knowledge. Still, our incompleteness result bears some significance for backward complete rules. For a rule $R : \Phi/\psi$ if Γ without auxiliary variables, i. e., $var(\psi) = var(\Phi)$, backward completeness implies completeness. Thus, as a consequence of Theorem 5, every sound and backward complete compositional circular A-G rule necessarily needs to employ auxiliary variables. In other words, backward reasoning with compositional circular rules is likely to require guessing auxiliary assertions about the system. I. e., one trades the lower complexity of the (decomposed) system for a higher complexity of the proof search.

6 Conclusion

We have shown that sound and compositional circular assume-guarantee rules, presented as inference rules restricted by an arbitrary side condition, cannot be complete. I. e., the side condition of a compositional rule, no matter how elaborate it is, cannot capture all cases where circular reasoning is admissible. Consequently, two important criteria for rating the quality of inference rules work against each other in the realm of circular reasoning. Upon designing assume-guarantee rules, this raises the question whether we should settle for compositionality or rather for completeness. The answer depends on the intended use of the rule.

Over the years, the practicality of circular assume-guarantee reasoning as a technique for compositional verification has been documented in a number of case studies, see [9, 10, 16] to name a few. In most cases, these assume-guarantee rules were tailored for model checking, and as model checkers particularly suffer from the infamous *state explosion* problem, the designers of the rules focussed on (automatic) system decomposition rather than on completeness. Consequently, these rules avoid to generate subgoals that involve a composition of subsystems. Here, compositional rules whose side conditions can be checked efficiently (but are not too restrictive) seem to be very appropriate. There are tools that successfully employ such incomplete compositional rules, e. g., in the verification of thread-parallel software [8]. To some extent, the loss of completeness can be mitigated against by human interaction, e. g., in the form of auxiliary annotations (to the code of the system), which provide more information about the system so the tools may find better decompositions.

To the best of our knowledge, there is no data available on the practical use of complete circular assume-guarantee rules in verification. However, in the case of manual (or almost manual) verification, we see no reason for severely restricting the power of circular reasoning, so one might prefer a complete rule over a compositional one. Of course, then one must tackle system decomposition in the subgoals by other means, e. g., by abstraction. Still, assume-guarantee reasoning may be superior to direct verification, as the additional assumptions in the subgoals may enable better abstractions.

References

[1] M. Abadi and L. Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, 1995.

[2] M. Abadi and S. Merz. An abstract account of composition. In *MFCS*, LNCS 969, pages 499–508. Springer, 1995.

[3] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.

[4] R. Alur and T. A. Henzinger. Reactive modules. In *LICS*, pages 207–218. IEEE Computer Society, 1996.

[5] B. A. Davey. Duality theory on ten dollars a day. In I. G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, volume 389 of *NATO ASI Series*, pages 71–111. Kluwer Academic Publishers, 1993.

[6] W. F. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[7] E. A. Emerson. Modal and temporal logics. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 135–191. Elsevier, 1990.

[8] C. Flangan, S. N. Freund, and S. Qadeer. Thread-modular verification for shared-memory programs. In *ESOP*, LNCS 2305, pages 262–277. Springer, 2002.

[9] T. A. Henzinger, X. Liu, S. Qadeer, and S. K. Rajamani. Formal specification and verification of a dataflow processor array. In *ICCAD*, pages 494–499. IEEE Computer Society, 1999.

[10] T. A. Henzinger, S. Qadeer, and S. K. Rajamani. You assume, we guarantee: Methodology and case studies. In *CAV*, LNCS 1427, pages 440–451. Springer, 1998.

[11] T. A. Henzinger, S. Qadeer, S. K. Rajamani, and S. Tasiran. An assume-guarantee rule for checking simulation. *ACM Transactions on Programming Languages and Systems*, 24(1):51–64, 2002.

[12] B. Jonsson and Y.-K. Tsay. Assumption/guarantee specifications in linear-time temporal logic. *Theoretical Computer Science*, 167(1–2):47–72, 1996.

[13] P. Maier. A set-theoretic framework for assume-guarantee reasoning. In *ICALP*, LNCS 2076, pages 821–834. Springer, 2001.

[14] P. Maier. *A Lattice-Theoretic Framework For Circular Assume-Guarantee Reasoning.* PhD thesis, Universität des Saarlandes, 2002. Submitted.

[15] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety.* Springer, 1995.

[16] K. L. McMillan. Verification of an implementation of Tomasulo's algorithm by compositional model checking. In *CAV*, LNCS 1427, pages 110–121. Springer, 1998.

[17] K. L. McMillan. Circular compositional reasoning about liveness. In *CHARME*, LNCS 1703, pages 342–345. Springer, 1999.

[18] J. Misra and K. M. Chandy. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, 7(4):417–426, 1981.

[19] K. S. Namjoshi and R. J. Trefler. On the completeness of compositional reasoning. In *CAV*, LNCS 1855, pages 139–153. Springer, 2000.

[20] V. Sofronie-Stokkermans. Representation theorems and the semantics of non-classical logics, and applications to automated theorem proving. In M. Fitting and E. Orlowska, editors, *Beyond two: Theory and Applications of Multiple-Valued Logic*, Studies in Fuzziness and Soft Computing 114, pages 59–100. Springer, 2003.

[21] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 135–191. Elsevier, 1990.

[22] M. Viswanathan and R. Viswanathan. Foundations for circular compositional reasoning. In *ICALP*, LNCS 2076, pages 835–847. Springer, 2001.

INFORMATIK

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-008	C. Rssl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Rssl, F. Zeilfelder, G. Nrnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nrnberger, C. Rssl, H.S. Seidel, F. Zeilfelder	The Dimension of $C^1 SplinesofArbitraryDegreeonaTetrahedralPartition$
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Rssl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	?
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Soon to be published
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vcking	Random Knapsack in Expected Polynomial Time

MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Trff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Trff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches
MPI-I-2001-1-006	T. Polzin, S. Vahdati	Partitioning Techniques for the Steiner Problem
MPI-I-2001-1-005	T. Polzin, S. Vahdati	On Steiner Trees and Minimum Spanning Trees in Hypergraphs
MPI-I-2001-1-004	S. Hert, M. Hoffmann, L. Kettner, S. Pion, M. Seel	An Adaptable and Extensible Geometry Kernel
MPI-I-2001-1-003	M. Seel	Implementation of Planar Nef Polyhedra
MPI-I-2001-1-002	U. Meyer	Directed Single-Source Shortest-Paths in Linear Average-Case Time
MPI-I-2001-1-001	P. Krysta	Approximating Minimum Size 1,2-Connected Networks

MPI-I-2000-4-003	S.W. Choi, H. Seidel	Hyperbolic Hausdorff Distance for Medial Axis Transform
MPI-I-2000-4-002	L.P. Kobbelt, S. Bischoff, K. Khler, R. Schneider, M. Botsch, C. Rssl, J. Vorsatz	Geometric Modeling Based on Polygonal Meshes
MPI-I-2000-4-001	J. Kautz, W. Heidrich, K. Daubert	Bump Map Shadows for OpenGL Rendering